



The PHP Company

User Guide

Zend Optimizer V3.3

By Zend Technologies, Inc.

Disclaimer

The information in this document is subject to change without notice and does not represent a commitment on the part of Zend Technologies Ltd. No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or information storage and retrieval systems, for any purpose other than the purchaser's personal use, without the written permission of Zend Technologies Ltd.

All trademarks mentioned in this document belong to their respective owners.

© 1999-2010 Zend Technologies Ltd. All rights reserved.

Zend Optimizer™ User Guide issued April 2010.

Product Version: Zend Optimizer 3.3

DN: OUG-T-250410-3.3-002

Table of Contents

Zend Optimizer™	4
Introduction	4
Overview	4
System Requirements	5
Installing the Zend Optimizer™	6
Uninstall	8
Installation Verification	8
Troubleshooting	9
Zend Optimizer™ Settings	12
Configure Optimizer Settings	12
Zend Optimizer PHP API Functions	13
Getting Additional Information	14
Zend Optimizer Technical FAQ	15

Zend Optimizer™

Introduction

The Zend Optimizer™ boosts PHP performance by going over the intermediate code generated by the standard Zend run-time compiler and optimizing it for faster execution. In addition, it runs the files encoded by the Zend Guard™, while enhancing the running speed of PHP applications.

The Zend Optimizer™ is available free of charge, from the Zend Website, <http://www.zend.com/en/products/guard/zend-optimizer>.

Overview

The Zend Optimizer™ is loaded as a PHP engine extension. This extension automatically optimizes PHP scripts, and transparently detects and loads encoded files.

The Zend Optimizer User Guide is intended for supplying information about installing and operating Zend Optimizer™ in the following categories:

- System Requirements
- Installation
- Troubleshooting
- Additional Information

System Requirements

In order to run properly, the following requirements have to be met:

1. Zend Optimizer™ is installed on a supported operating system.
2. Zend Optimizer™ is installed on a supported version of PHP 4 or PHP 5.
3. A Web Server is installed on the machine on which the PHP is running.

A complete list of system requirements for the Zend Optimizer™ is constantly kept up to date with new versions, updates and changes in the System Requirements section on the [Zend Website](#):

<http://www.zend.com/en/products/guard/system-requirements>

Installing the Zend Optimizer™

This section describes the Zend Optimizer™ installation procedures for all compatible operating systems.

Important note:

To install or uninstall Zend Optimizer™, you must be logged in with administrator/root permissions.

The Zend Optimizer™ is a free package that can be downloaded from the Zend Website.

To download and install the Zend Optimizer™:

1. Login or register with your Zend account and go to the Downloads tab.
2. Click the Download button for the Zend Optimizer™ product.
The Zend Optimizer™ – Free Download page opens
(<http://www.zend.com/en/products/guard/zend-optimizer>).
3. Select the Zend Optimizer™ version that corresponds with the platform system you use and select “Download Now”.
The File Download dialog opens.
4. Choose the Save option and save the selected file to your local system.

From this point on the installation instructions differ according to the operating system on which you would like to install the Zend Optimizer™.

To install the Zend Optimizer™ on Windows:

1. Double click on the package ZendOptimizer-x.x.x-windows-i386.exe
The Optimizer Install Shield opens.
2. Follow the instructions in the installer.
These instructions define the basic system configurations necessary for installing the Zend Optimizer™ such as the Web Server in use and the location of the php.ini.

You can now jump to the section: Installation Verification (page 8), for instructions on how to confirm that the Zend Optimizer™ was successfully installed.

To install the Zend Optimizer™ on UNIX Variants and Mac OS X:

1. Open a terminal window and change dir to the directory in which the Zend Optimizer™ package was saved.
To untar the package use the command:
`gunzip -c <package name> | tar xf -`
2. Take the .so file that corresponds to your PHP version and copy it into your extension directory.
3. In your php.ini load the .so file using the following directive (where “/usr/local/zend/lib” is your directory and “zendoptimizer.so” is your .so file):
`Zend_extension=/usr/local/zend/lib/ZendOptimizer.so`

Zend Optimizer is now installed. Before using Zend Optimizer™, restart your PHP.

Note:

After installing the Zend Optimizer™, check whether the installation was successfully completed (see Checking for Successful Installation.)

Uninstall

Important note:

To install or uninstall Zend Optimizer™, you must be logged in with administrator/root permissions.

Windows

In order to uninstall Zend Optimizer™:

1. Double click on the uninstaller icon, which is located at Control Panel | Add/Remove Programs and select Zend Optimizer.

UNIX Variants and Mac OS X

In order to uninstall the Zend Optimizer™:

1. Replace the php.ini file with the backed up copy (php.ini-zend_optimizer.bak).
2. Delete the Zend Optimizer™ installation directory.

Important note:

If you have other Zend products installed under the same directory, do not delete the installation directory. In this case, you should only change the php.ini file.

3. Restart the Web Server.

Installation Verification

This program makes use of the Zend scripting language engine.

Use the PHP function: `phpinfo()` to verify a successful installation, as shown in the following script:

```
<?php
    phpinfo();
?>
```

The part of the output where the Zend Optimizer™ is listed looks similar to the following:

```
Zend Engine vX.Y.Z, copyright (c) 1998-2010, Zend Technologies Ltd.,
with the Zend Optimizer vX.Y.Z, copyright (c) 1998-2010 Zend Technologies
Ltd.
```


Troubleshooting

This section discusses the possible incompatibilities that may arise and how they can be prevented in order to properly activate and use the Zend Optimizer™.

Limiting the Number of Concurrent Licenses and “Cookies”

When the “Require Cookies” option is **NOT** selected, any page can be loaded, once, from any number of browsers concurrently even when the number of concurrent users is limited (even to a single user).

This happens because – the first time that the page is accessed, no cookie is present. The second time that the page is accessed, a cookie *is* present (even though not required). The server then limits access.

In addition, if the “Require Cookies” option is **NOT** selected, AND the browser is set to not receive cookies, any page can be loaded, any number of times, from any number of browsers.

This happens because no cookie is present at any time. The server has no way to know that the page has been accessed.

PHP Version Incompatibility

The Zend Optimizer™ must be fully compatible with the PHP version installed. An incompatibility will cause the Zend Optimizer™ to fail and register an error message. (See message in the Web Server log file). Make sure the file you downloaded corresponds with the installed PHP version. If necessary, uninstall the Zend Optimizer™ and download/install the version suitable for the PHP version in use.

Note:

Zend Optimizer™ 3.3 supports all recent PHP 4 and PHP 5 versions, not including non-released versions and old PHP versions (like 4.0.x).

Note:

The Zend Optimizer™ only runs with supported Web Servers. See the [Zend Optimizer System Requirements \(http://www.zend.com/en/products/guard/system-requirements\)](http://www.zend.com/en/products/guard/system-requirements) section of the Zend Website for a list of supported Web Servers.

Configuration Options

The Zend Optimizer™ will not run if the following PHP settings are enabled:

- **Debug Mode** - Check that your PHP was compiled in non-debug mode. This can be checked by looking at the top part of the phpinfo() function output and checking "Debug Build" row (ZEND_DEBUG in older PHP versions). It should say "no".
- **Thread Safety** - Check that your PHP was compiled in ZTS mode for Windows and non-ZTS mode for UNIX. This can be checked by looking at the top part of the phpinfo() function and checking "Thread Safety" row. It should say "disabled" for Unix and "enabled" for Windows.
- **Enable Versioning** - Check that your PHP was compiled without the Enable Versioning setting. This can be checked by looking at the top part of the phpinfo() function and checking to see if a row called "--enable-versioning" exists in the Configure Command section. If it does exist the PHP should be compiled without this option.

Other PHP Extension Incompatibilities

Another reason for the Zend Optimizer™ not running could be due to extension incompatibility. This problem arises when the php.ini includes extensions from different sources that interfere or clash with the Zend Optimizer™. As a general rule, any extension that is related to debugging or optimizing that has been added to the php.ini by a third party supplier, has the potential to disrupt the performance of the Zend Optimizer™ should be disabled/removed.

Zend Guard™ Incompatibility

Files encoded with the Zend Guard™ must be compiled with a compatible version of Zend Guard™.

When in doubt always use the latest version of Zend Optimizer™.

If the versions of Zend Guard™ and Zend Optimizer™ are incompatible, the Zend Optimizer™ will fail to run encoded files and will register an error message (“Zend Optimizer not installed” or “General Optimizer error”).

Note:

If you are using Zend Guard™ 4.0 and above, use the latest Zend Optimizer™ version.

Zend Optimizer™ Settings

As mentioned earlier, the Zend Optimizer™ goes over the intermediate code generated by the standard Zend run-time compiler and optimizes it for faster execution.

In order to customize the Optimizer's performance, the following directives and extensions have been provided.

Configure Optimizer Settings

Directive	Description
zend_optimizer.enable_loader	If you do not plan to use the Zend Optimizer™ to load encoded files, you can slightly improve the Optimizer's performance by adding the <code>zend_optimizer.enable_loader = 0</code> . This disables the transparent auto-loading mechanism that is built into the Zend Optimizer™.
zend_optimizer.disable_licensing	If you do not need to use any licensing features, you can disable the Zend Optimizer™ license request. Setting this option lowers Optimizer memory usage and slightly enhances performance.
zend_optimizer.license_path	A license file is required to load encoded PHP scripts that were encoded with a "license" option. If you turn off this option, the encoded scripts on your server that require a license may not load. This directive is a list of directories or files, separated by path separator (";" on windows, ":" on Unix) that gives the Optimizer a place to look for encoded file licenses. If a .zl file that was added to this path was loaded as a license it will be taken individually. If instead of a file there is a directory, all .zl files in that directory will be loaded as licenses. Licenses are loaded on server start time (unless loaded by PHP API), so if you add a license or change a path, you should restart the server.
zend_optimizer.obfuscation_level_support	The Obfuscation level supported by the Zend Optimizer™.

Located in the zend.ini, these directives can be easily configured through Zend Platform and Zend Core (all flavors).

Note:

If you are using the Zend Optimizer™ along with Zend Guard™ to encode files, please refer to the [Zend Guard User Guide](http://files.zend.com/help/Zend-Guard/zend-guard.htm) (<http://files.zend.com/help/Zend-Guard/zend-guard.htm>) for more information.

Zend Optimizer PHP API Functions

The following table lists the PHP API functions that can be used with the Zend Optimizer™:

Function	Syntax	Description
zend_loader_enabled	bool zend_loader_enabled()	Shown if loader is enabled.
zend_loader_file_encoded	bool zend_loader_file_encoded()	Returns true if the current file is a Zend-encoded file.
zend_loader_file_licensed	array zend_loader_file_licensed()	Returns license (array with fields) if the current file has a valid license and is encoded, otherwise it returns false.
zend_loader_current_file	string zend_loader_current_file()	Returns the name of the file currently being executed.
zend_loader_install_license	bool zend_loader_install_license(string license_file[, bool override])	Dynamically loads a license for applications encoded with ZendGuard™. The Override controls if it will override old licenses for the same product.
obfuscate_function_name	string obfuscate_function_name(string function_name)	Obfuscate and return the given function name with the internal obfuscation function.
obfuscate_class_name	string obfuscate_class_name(string class_name)	Obfuscate and return the given class name with the internal obfuscation function.
current_obfuscation_level	int current_obfuscation_level()	Returns the current runtime obfuscation level.

Note:

To insure backward compatibility with scripts obfuscated with the Zend Guard™ at level "Basic", their runtime obfuscation level will remain 0 !

zend_runtime_obfuscate	bool zend_runtime_obfuscate()	Start runtime-obfuscation support that allows limited mixing of obfuscated and un-obfuscated code.
zend_get_id	array zend_get_id([bool all_ids = false])	Returns array of the host ids. If all_ids is true, then all IDs are returned, otherwise only IDs considered "primary" are returned.

Getting Additional Information

There are several sources of information that Zend Optimizer™ users can use in order to find answers to questions that may arise.

Answers to technical and operating system related questions can be found in the Zend Knowledge Base. Here you will find a repository of articles that address and manage known issues:

<http://kb.zend.com/>

Otherwise, you can access one of the Zend forums at the Zend Support Center:

<http://www.zend.com/support/>

For issues other than technical support, email us at: support@zend.com

Zend Optimizer Technical FAQ

What is the Zend Optimizer™? What does it do?

The Zend Optimizer™ goes over the intermediate code generated by the standard Zend run-time compiler and optimizes it for faster execution. In addition, the Zend Optimizer™ enables PHP to load and execute files encoded by the Zend Guard™.

How do I get the Zend Optimizer™?

The Zend Optimizer™ is available for download free of charge from the [Zend Products Download Center](http://www.zend.com/downloads/), at <http://www.zend.com/downloads/>.

I have installed the Zend Optimizer™. How do I start it?

The Zend Optimizer™ is a passive module that runs within the framework of PHP. It optimizes scripts automatically, and transparently detects and loads encoded files, therefore eliminating the need for you to start it.

How many optimizations does the Zend Optimizer™ perform altogether?

The Zend Optimizer™ is capable of performing around 20 different optimizations, some of which are quite complex. One of the simpler optimizations of the Zend Optimizer™ is to change, where possible, post-incrementing to pre-incrementing, since pre-incrementing is the faster operation of the two.

Is there any way to further improve the performance of the Zend Optimizer™?

If you don't plan to use the Zend Optimizer™ in order to load encoded files, you can slightly improve the performance of the Optimizer by adding the `zend_optimizer.enable_loader = 0` directive to `php.ini`. This disables the transparent auto-loading mechanism that is built into the Zend Optimizer™.

Can encoded and non-encoded (source) PHP files be used together?

Encoded and non-encoded PHP files can be used together transparently.

Will using encoded files (instead of source files) change run-time speed or file size?

There might be some speed gain because the compilation stage is saved on every run of the script. However, speed improvements, if any, are dependent on the nature of the script. The size of an encoded file might be somewhat smaller or larger than the source file, but this too is dependent on the nature of the script. The factors that tend to improve run-time speed are not necessarily the same as those that tend to decrease file size.

When I start my server, the Zend Optimizer gives error messages and fails: Why?

Your version of PHP may be incompatible with the Zend Optimizer that you installed. Make sure that you download a version of the Zend Optimizer™ that is compatible with your version of PHP.

When I try to run an encoded file, error messages are displayed: Why?

The most common cause of error messages and failure is incompatibility either with the PHP version, or with the Zend Intermediate Code file (that is, with the version of the Zend Guard™ that encoded the file). Another possible cause for failure could be encoded file corruption caused by transferring the file between systems. This happens because even though encoded files are system-independent, certain tools for transferring files from Windows to Unix and vice-versa change the contents of the files so that they appear to be text files. To prevent this from happening, always use "binary transfer mode" when transferring encoded files with file transfer tools or ensure integrity of the files by using a package such as: zip or tar.gz.

For additional information, see the "Avoiding Incompatibilities" section in the Zend Guard User Guide.